Here's what I got:

I attached the function I used to output this. You just need to define "cym" in the top of the file.

```
0: #FFFFFF
1: #FEF6FE
2: #FEEFEE
3: #FEE7FD
4: #FEDFFC
5: #FED7FB
6: #FDCFF9
7: #FDC7F8
8: #FCBFF6
9: #FCB7F3
10: #FBB0F1
11: #FBA8EE
12: #FAA1EB
13: #F999E7
14: #F892E4
15: #F78BE0
16: #F684DC
17: #F57DD8
18: #F476D3
19: #F36FCF
20: #F269CA
21: #F162C5
22: #EF5CC0
23: #EE56BB
24: #ED50B5
25: #EB4AB0
26: #EA45AA
27: #E83FA4
28: #E63A9F
29: #E43599
30: #E33093
31: #E12C8D
32: #DF2787
33: #DD2381
34: #DB1F7B
35: #D91B75
36: #D7186F
37: #D51469
38: #D21163
39: #D00F5D
```
If anyone hard codes anything I will give them a 0.

Does that mean that my output is good?

Idk, didn't look. I will let Dr. Lusth be the decider on who can respond.

The hive mind needs to approve or disapprove.
My output matches

This is why dealing with floating point is so tricky. In one case, one of the intermediate results cannot be represented exactly in floating point.

```c
#include <stdio.h>
#include <math.h>

#define pi 3.14159265358979323846

int main(void)
{
    printf("%d\n", (int) (255 * (1 - sin((pi * 0.01) * 100))));
    printf("%d\n", (int) (255 * (1 - sin(pi * (0.01 * 100)))))
    return 0;
}
```

Since Scam uses C as its host language, a Scam version of this program produces the same results.

Gnuplot is a system for plotting equations. Here is a version for plotting a sin wave that is "close" to what you will need. I basically played with the equation until I got it to look like I wanted. Then I translated the equation to Scheme. Here are instructions for Ubuntu:

Place the following code in a file named plot.gplot:

```bash
#!/usr/bin/gnuplot
set term postscript enhanced eps monochrome
set output "plot.eps"
set title "Sin Wave"
set xlabel "Input"
set ylabel "Output"
set pointsize 1.5
```

Since Scam uses C as its host language, a Scam version of this program produces the same results.
#set key spacing 2
set nokey
set xrange [0:100]
set yrange [-255:255]
#set xtics 0,2500
#set ytics 0,250
plot 255 * sin((2 * pi * x) / 20)

Install gnuplot if you haven't done so already:

```bash
sudo apt-get install gnuplot
```

Then run the command:

```bash
gnuplot plot.gplot
```

Then look at the result:

```bash
evince plot.eps &
```

You can rerun gnuplot and see the result in the updated evince window.

---

Subject: Re: Problem 3 Output - Let's Compare
Posted by eadwyer on Thu, 01 Sep 2016 03:57:11 GMT

View Forum Message <> Reply to Message

I concur with these results